REMARKS

Reconsideration of the application in view of the above amendments and the following remarks is respectfully requested. Claims 1-20 and 31-40 have been canceled. These claims will be pursued in one or more divisional applications. No claims have been amended or added. Claims 21-30 are currently pending in the application.

In the Office Action, the Examiner objected to the Declaration as lacking the signatures of some of the inventors. Applicants respectfully submit that this objection was made in error. In the papers filed on January 5, 2001, four Declarations were provided. Each Declaration was signed by one of the four inventors. Therefore, all of the inventors executed a Declaration. A copy of the papers filed on January 5, 2001, is enclosed herewith to support this statement. Accordingly, Applicants request that this objection be withdrawn.

In the Office Action, the Examiner objected to the drawings for including some reference signs that were not mentioned in the Specification. The Examiner required either: (1) that the drawings be amended to remove the reference signs; or (2) that the Specification be amended to include mention of the reference signs. Applicants have amended the Specification to include mention of all of the reference signs noted by the Examiner. Accordingly, Applicants request that this objection be withdrawn. The Examiner is thanked for his careful review of the application.

Claim Rejections – 35 U.S.C. §102(e)

In the Office Action, the Examiner rejected Claims 21, 22, and 26-29 under 35 U.S.C. § 102(e) as being anticipated by Berkowitz (U.S. Patent No. 6,529,921). This rejection is respectfully traversed.

Independent claim 21

With regard to independent claim 21, there is recited:

> A computer-implemented method for performing a transaction comprising the steps of:
> producing a transaction instance data structure indicating a plurality of operations constituting a transaction; the transaction instance data structure indicating a linking of the plurality of operations to indicate an operation performance order; the transaction instance data structure further indicating conditioning logic data for changing the operation performance order such that the plurality of operations is capable of being performed in more than one possible order; and
> for each of the plurality of operations,
> producing an operation request message indicating input data for performing an operation;
> sending the operation request message to a service application to perform the operation using the input data;
> receiving an operation response message from the service application indicating output data from the operation; and
> determining a next operation to perform using the conditioning logic data and the output data of the operation response message.

Claim 21 provides an advantageous method for managing the performance of a transaction. The method may be applied in many contexts, including one in which a transaction is carried out in a distributed network system.

According to claim 21, when it is desired to perform a transaction, a transaction instance data structure is produced. This data structure indicates a plurality of operations constituting the transaction. The data structure also indicates a linking of the operations to indicate an operation performance order, and further indicates conditioning logic data for changing the performance order of the operations. Basically, the data structure acts as a specification for the transaction.

After the data structure is produced, a number of steps are carried out for each of the operations indicated in the data structure. These steps include producing an operation request message indicating input data, sending the operation request message to a service application to perform the operation using the input data, receiving an operation response message from the service application indicating output data from the operation, and determining a next operation to perform using the conditioning logic data in the data structure and the output data of the operation response message. By performing these steps for each operation in the transaction, the overall transaction is carried out.

The method of claim 21 provides many different advantages. One such advantage is that it enables the entire transaction to be managed from a central point. Even if the operations of the transaction are performed by service applications situated at different nodes in a distributed network system, it does not matter. The method of claim 21 can still be used to manage the transaction. As a result, the method of claim 21 makes it possible to manage a transaction centrally and conveniently.

Such a method is neither disclosed nor suggested by Berkowitz. Instead, Berkowitz discloses a method and apparatus for maintaining a coherent set of database tables (i.e. a coherent cache) among a plurality of nodes. In Berkowitz, when a new node joins the plurality of nodes (and hence, joins the "coherent cache"), one of the existing nodes is selected to be a source node. The task of the source node is to provide the new node with a set of database tables that are current and synchronized with the tables on all of the other nodes.

To do so, the source node first provides the destination node with the data that is currently in its database tables. While this data is being provided, the database tables may be updated by ongoing transactions. Thus, the source node also provides to the new node a "snapshot", which captures all of the active transactions (transactions that have not been committed or aborted) that involve the tables being synchronized. The new node can thereafter use the snapshot information to update the information in its tables, as needed. The new node is thus brought online into the coherent cache.

In rejecting claim 21, the Examiner interpreted (by citing Col. 14, lines 30-65 of Berkowitz) the snapshot of Berkowitz to be the "transaction instance data structure" recited in claim 21, and the changes included in the snapshot to be the "operations" recited in claim 21. In light of this interpretation, a number of points should be noted.

First, it should be noted that the changes in the snapshot do not truly constitute "operations" in that they do not call for any action to be performed. Rather, they are just sets of data that are provided to the new node. The new node may at some point decide to use the sets of data to update its tables, but the sets of data in and of themselves do not call for any action to be performed. Thus, Applicants submit that the changes included in the snapshot cannot be fairly read to be the "operations" recited in claim 21.

Another point to note is that, unlike the transaction instance data structure of claim 21, the snapshot of Berkowitz does not indicate any linking between the changes to indicate a performance order. More particularly, there is no relationship specified between the changes to indicate any type of order between the changes. As far as the source node is concerned, the changes may be sent to the new node in any order. In support of the rejection, the Examiner cites Col. 6, lines 15-40 of Berkowitz, which discusses the arbitration process implemented by the nodes to determine an order in which to apply changes. Applicants respectfully submit that the portion cited by the Examiner has little to do with the snapshot process. In fact, as stated in Col. 8, lines 20-21, arbitration is frozen during the creation of the snapshot. Thus, none of the discussion provided in Col. 6, lines 15-40 pertaining to arbitration relates to the snapshot process. Overall, there is nothing in the snapshot process that indicates a linking of the changes to indicate a performance order.

Yet another point to note is that, unlike the transaction instance data structure of claim 21, the snapshot of Berkowitz has no conditioning logic data that can be used to alter the performance order of the changes. As noted above, the snapshot has no linking information to indicate a performance order. Thus, it follows that the snapshot also has no conditioning logic data that can be used to change the performance order. In support of the rejection, the Examine contends that the status of a transaction (commit or abort) can be read to be the conditioning logic data. Applicants respectfully disagree. Whether a transaction has been committed or aborted (i.e. whether the transaction is still active) determines whether a change is included in the snapshot. It does not determine, nor can it be used to change the order in which the changes are sent to the new node. Berkowitz does not disclose or suggest anything in the snapshot that can be fairly read to be conditioning logic data.

Yet another point to note is that, unlike claim 21, when the source node sends a change to the new node, the source node is not sending an "operation request" to the new node. Put another way, the source node is not asking the new node to perform any operation. Rather, it is simply sending the new node a set of data. Further, unlike claim 21, the source node does not receive anything from the new node. It certainly does not receive anything that even remotely resembles output data from an operation, as recited in claim 21. Since the source node is not asking the new node to perform any operation, it should come as no surprise that the source node does not receive any output data from the new node. Further, unlike claim 21, the source node does not determine a next operation to perform using the conditioning logic data and the output data. As noted above, the snapshot has no conditioning logic data, and the source node receives no output data from the new node. Thus, the source node cannot possibly determine a next operation using conditioning logic data and output data. In support of the rejection, the Examiner cites various portions of Berkowitz that do not relate to the snapshot process. Applicants submit that these portions cannot fairly be read into the snapshot process to reject claim 21.

As argued above, Berkowitz fails to disclose or suggest a number of aspects of claim 21. Accordingly, Applicants submit that claim 21 is patentable over Berkowitz.

Applicants further submit that dependent claims 22 and 26-27, which depend from claim 21 and which recite further advantageous aspects of the invention, are likewise patentable over Berkowitz for at least the reasons given above in connection with claim 21.

Claim 28 is an article of manufacture claim which is analogous to the method of claim 21. Applicants submit that claim 28 is patentable over Berkowitz for at least the reasons given above in connection with claim 21.

Applicants further submit that dependent claim 29, which depends from claim 28 and which recites further advantageous aspects of the invention, is likewise patentable over Berkowitz for at least the reasons given above in connection with claim 28.

Claim Rejections – 35 U.S.C. §103(a)

In the Office Action, the Examiner rejected claim 23 under 35 U.S.C. § 103(a) as being unpatentable over Berkowitz in view of Chen (U.S. Patent No. 6,507,856). This rejection is respectfully traversed.

The Examiner relies on Chen to show request and response messages that include XML tags. For the sake of argument, it will be assumed that Chen discloses the subject matter as contended by the Examiner. Even if this were true, however, the combination of Berkowitz and Chen still would not produce the method of claim 23. As argued above in connection with claim 21 (from which claim 23 depends), Berkowitz fails to disclose or suggest a number of aspects of claim 21. The combination of Berkowitz and Chen would still have these shortcomings. As a result, even if the references were combined, they still would not disclose every element of claim 23. Thus, Applicants submit that claim 23 is patentable over Berkowitz and Chen.

In the Office Action, the Examiner rejected claims 24, 25 and 30 under 35 U.S.C. §103(a) as being unpatentable over Berkowitz in view of Srinivasan (U.S. Patent No. 5,893,108). This rejection is respectfully traversed.

The Examiner relies on Srinivasan to show the use of a directed acyclic graph (DAG). For the sake of argument, it will be assumed that Srinivasan discloses the subject matter as contended by the Examiner. Even if this were true, however, the combination of Berkowitz and Srinivasan still would not produce the method of claims 24 and 25. As argued above in connection with claim 21 (from which claims 24 and 25 depend), Berkowitz fails to disclose or suggest a number of aspects of claim 21. The combination of Berkowitz and Srinivasan would still have these shortcomings. As a result, even if the references were combined, they still would not disclose every element of claims 24 and 25. Thus, Applicants submit that claims 24 and 25 are patentable over Berkowitz and Srinivasan.

Claim 30 is an article of manufacture claim which is analogous to the method of claim 24. Applicants submit that claim 30 is patentable over Berkowitz and Srinivasan for at least the reasons given above in connection with claim 24.

For the reasons given above, Applicants submit that the pending claims are patentable over the art of record, including the art cited but not applied. Accordingly, allowance of all pending claims is respectfully solicited.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: 10/31, 2003

Bobby K. Truong
Reg. No. 37,499

1600 Willow Street
San Jose, California 95125-5106
Telephone No.: (408) 414-1080
Facsimile No.: (408) 414-1076

Attachments

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Non-Fee Amend, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

on Oct. 31, 2003 (Date) by (Signature)